# Curvature-based Sparse Rule Base Generation for Fuzzy Interpolation using Menger Curvature

Zheming Zuo[1], Jie Li[2], and Longzhi Yang[1]

[1] Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K.
{zheming.zuo, longzhi.yang}@northumbria.ac.uk
[2] School of Computing & Digital Technologies, Teesside University, Middlesbrough, U.K.
jie.li@tees.ac.uk

**Abstract.** Fuzzy interpolation improves the applicability of fuzzy inference by allowing the utilisation of sparse rule bases. Curvature-based rule base generation approach has been recently proposed to support fuzzy interpolation. Despite the ability to directly generating sparse rule bases from data, the approach often suffers from the high dimensionality of complex inference problems. In this work, a different curvature calculation approach, i.e., the Menger approach, is employed to the curvature-based rule base generation approach in an effort to address the limitation. The experimental results confirm better efficiency and efficacy of the proposed method in generating rule bases on high-dimensional datasets.

**Keywords:** Fuzzy Interpolation, Rule Base Generation, Sparse Rule Base, Menger Curvature, High-dimensional Data

## 1 Introduction

Fuzzy inference systems are built upon the fuzzy sets and fuzzy logic theory to provide a mapping mechanism that maps system input spaces to output spaces. A typical fuzzy inference system consists of two components: a rule base and an inference engine. A number of inference engines have been proposed, with the Mamdani inference and the TSK inference being the most widely applied. The TSK fuzzy inference is able to produce crisp outputs directly, as the polynomials are employed in the rule consequences. In contrast, the Mamdani fuzzy model is more intuitive and suitable for coping with linguistic inputs, thereby, to produce the fuzzy outputs. Common to both fuzzy inference approaches, Mamdani and TSK approach, a dense rule base, in which the entire input domain is fully covered, is required to support the fuzzy inference.

A fuzzy rule base can usually be generated in one of two ways: knowledge-driven, which generates fuzzy rule bases from expert knowledge, or data-driven that extracts rule bases from existing data. The knowledge-driven approaches essentially are a representation of the human expertise in the format of fuzzy rules that require full understanding of the problems by human experts. Recognising that the expert knowledge may not always be available, data-driven approaches were proposed, which extract fuzzy rules from a set of training data. Such data-driven approaches are commonly built upon

a large quantity of existing data to target the dense rule bases used by the conventional fuzzy inference engines.

Fuzzy rule interpolation (FRI), which was initially proposed in [1], relaxes the requirement of dense rule bases from conventional fuzzy inference systems [1]. When a given input does not overlap with any rule antecedent, certain conclusions can still be obtained by means of interpolation. A number of fuzzy interpolation approaches have been developed, such as [1–8], which have been successfully applied to deal with real-world problems, such as cybersecurity [9–11], smart home control system [12], network QoS management system [13], personalised exoskeleton control [14], and job planning system [15]. Fuzzy interpolation is not only able to enhance fuzzy inference over a sparse rule base but also able to help in system complexity reduction by removing the rules that can be approximated by others [16]. To reduce the complexity of such rule bases, various rule base reduction approaches have been developed [16–18]. Nevertheless, those approaches usually generate a dense rule base first, which is followed by the removing of redundancy rules based on certain similarity measures. As a consequence, such operations are likely to lead to an extra computational cost.

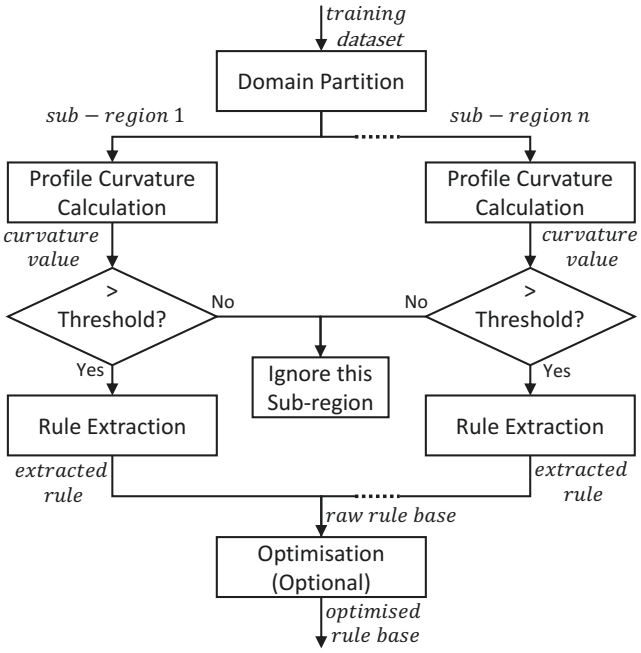
This paper proposes a different curvature-based data-driven rule base generation approach [19, 20] for FRI which not only directly generates sparse rule bases from the given training datasets but also copes well with high-dimensional data. Fundamentally, the majority of existing fuzzy interpolation approaches are fuzzy extensions of crisp linear interpolation. Based on this, the 'flat' or 'straight' regions of a data pattern can be easily approximated by its surroundings. The curvature value of a part of the data pattern in a way indicates the straightness of the given region of the data, where the lower curvature value represents the flatness and vice verse. Based on this observation, the proposed system first determines and removes the 'flat' or 'straight' regions of the given data by calculating the curvature values, thus only extracts rules from the selected regions that have high-curvature values. The proposed approach is evaluated by two experiments with promising results generated.

The rest of the paper is structured as follows: Section 2 revisits the related background theories, including the Transformation-based fuzzy interpolation approach and the conventional curvature-based rule base generation approach. Section 3 presents the proposed system. Section 4 demonstrates and evaluates the proposed system. Section 5 concludes the paper and suggests probable future developments.

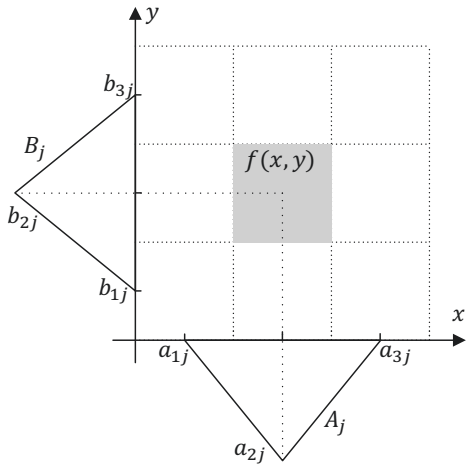
## 2 Background

Recent development of rule base generation has been reported, with compact sparse rule bases targeted [20]. This approach is developed based on the concept of profile curvature values of different parts of the data pattern. In particular, the method divides the problem domain into a number of components (sub-regions). As the 'flat' or 'straight' parts of the pattern can be approximated by linear fuzzy interpolation techniques, only the parts with higher curvature values are selected for fuzzy rules extraction. The approach is illustrated in Fig. 1(a).

*Domain partition:* Given a training dataset, which is distributed in a 3-dimensional space (2-inputs and signal output), the input domain is equally partitioned into $a \times b$




(a) The work flow of the curvature-based rule base generation




(b) Fuzzy sets extraction

Fig. 1: Curvature-based rule base generation

grid areas, where $a$ and $b \in \mathbb{N}$ indicate the number of partitions on a horizontal axis and a vertical axis, respectively.

*Curvature value calculation:* The profile curvature values are usually used in geospatial analysis, which represents the steepest downward gradient for a given direction [21]. The profile curvature values are used in this approach to help indicate the importance of each sub-region. Given a sub-region $f(x, y)$, the profile curvature, $k_p$, is the rate at which a surface slope, $S$, changes whilst moving in the direction of $grad(f)$, which can be calculated by the directional derivative:

$$D_{(\hat{n})}(F) = \bigtriangledown F \cdot \hat{n}. \tag{1}$$

The directional derivative refers to the rate at which any given scalar field, $F(x, y)$, is changing as it moves in the direction of some unit vector, $\hat{n}$, such as $\hat{n} = -(\bigtriangledown f/S)$, where $S$ is the slope defined as the magnitude of the gradient vector and is a scalar field:

$$S(x, y) = |\bigtriangledown f| = \sqrt{f_x^2 + f_x^2}, \tag{2}$$

where $\bigtriangledown f = (f_x, f_y, 0)$ denotes the gradient of this surface, which is a 2D vector that points in the steepest uphill and downhill directions. From here, the profile curvature values, $k_p$, can be expressed as:

$$k_p = -S^{-1}(\bigtriangledown S \cdot \bigtriangledown f), \tag{3}$$

In order to calculate the overall linearity of a sub-region, eight directions of profile curvature values, which are defined from the centre of the sub-region to the four corners and the central points of the four edges, are defined, such as $k_{p_i}, i = \{1, 2, \cdots, 8\}$. That is, the final profile curvature value takes the maximum value of the eight directional curvature values: $k_p = \max(k_{p_i}),\ i = \{1, 2, \cdots, 8\}$.

*Rule extraction:* Given a curvature threshold $\theta$, if the curvature value of a sub-region is greater than $\theta$, the corresponding sub-region will be selected to form a fuzzy rule. In this approach, each selected sub-region is represented by one fuzzy rule. For simplicity, only isosceles triangular fuzzy sets are employed in this approach, each of which can be precisely represented as $A = (a_1, a_2, a_3)$, where $a_2$ is the core and $(a_1, a_3)$ is the support of the fuzzy set. In this approach, the core of the fuzzy set is set to the centre of the sub-region, and the support of the fuzzy set is equal to twice the span of the corresponding sub-region. Given a selected sub-region $f(x, y)$, the extracted fuzzy sets are illustrated in Fig. 1(b). The raw rule base can then be constructed from all extracted rules.

*Optimisation:* The generated raw rule base can be employed for results generation. An optional process, the rule base optimisation, can also be applied in the end to fine-tune the parameters by adopting a genetic optimisation algorithm, such as a Genetic Algorithm (GA) [22], in order to increase the performance. The details of the optimisation process are omitted here, as it is not the main focus of this paper.

The profile curvature is based on the directional derivative, which usually used in 3D surface. As a consequence, the profile curvature-based rule base generation approach is limited on three-dimensional problems, which comprise two inputs and signal output.

## 3 Menger Curvature-based Rule Base Generation

The original curvature-base rule base generation approach as introduced in Section 2 is extended in this section by deploying the concept of the Menger Curvature [23], thereby to relief the limitation of the original approach by allowing to handle the high-dimensional data instances.

### 3.1 Menger Curvature

The Menger Curvature (MC) measures the curvature of a triple of points in n-dimensional Euclidean space $\mathbb{E}^n$ which is the reciprocal of the radius of the circle that passes through the three points [23]. In this work, only plane curves, that is, only two-dimensional problems, are considered. Assume that $p_1(x_1, y_1), p_2(x_2, y_2), p_3(x_3, y_3)$ are three points in a two-dimensional space $\mathbb{E}^2$ and $p_1, p_2, p_3$ are not collinear, as depicted in Fig. 2, the MC on $p_2$ can be defined by:

$$MC(p_1, p_2, p_3) = \frac{1}{R} = \frac{2\sin(\varphi)}{||p_1, p_3||}, \tag{4}$$

where $R$ represents the $||p_1, p_3||$ denoting the Euclidean distance between $p_1$ and $p_3$, and $\varphi$ is the angle made at the $p_2$-corner of the triangle spanned by $p_1, p_2, p_3$, which can be obtained by the Law of Cosines:

$$cos(\varphi) = \frac{||p_1, p_2||^2 + ||p_2, p_3||^2 - ||p_1, p_3||^2}{2 \cdot ||p_1, p_2||^2 \cdot ||p_2, p_3||^2}. \tag{5}$$

Note that the MC on point $p_1$ and $p_3$ will not be calculabled, due to the boundary points.

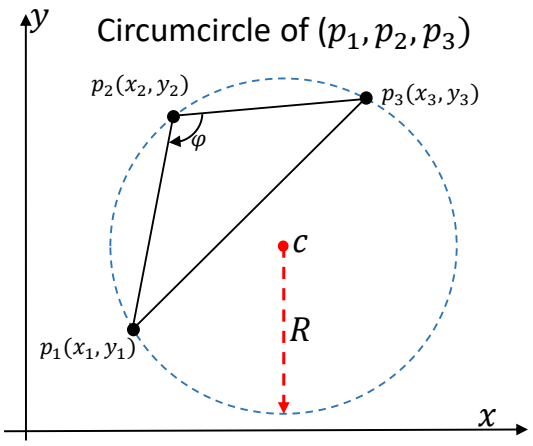



Fig. 2: The Menger Curvature of a triple of points on two-dimensional space

### 3.2 Extended Rule Base Generation

Given a high-dimensional complex inference problem, the same procedure as detailed in Sec. 2 is used for rule base generation, except the step of curvature value calculation. In particular, a high-dimensional data instance is first evenly partitioned to a number of hypercubes. Then, each hypercube is broken down into a set of two-dimensional (2D) problems with one input and a single output. From there, the MC is applied to each data point of the obtained 2D problems, thus to calculate the mean of MC of each 2D problem. Finally, the curvature of each generated hypercube can be determined by the weighted average of the curvature values of its corresponding 2D problems. The hypercubes with higher curvature values are then selected to contribute to the fuzzy rule base generation.

Given a high-dimensional problem denoted as $\mathbb{P}_{n+1}(n > 2)$, assume it contains $n$ inputs features $\mathbb{X} = \{x_1, \ldots, x_n\}$, and single output feature $y$, which have been evenly partitioned into $m_1 \times \cdots \times m_n$ hypercubes, where $m_i(1 \leqslant i \leqslant n)$ denotes the number of partitions in the input domain of $x_i$. Taking a hypercube $H_i$, which contains $h_i$ data instance, as an example, its curvature value can be computed by the following steps:

**Step 1** *Hypercube break down*: Given a hypercube $H_i$, which contains $n$ input features and single output feature, it is first broken down into $n$ 2D planes, denoted as $P_j^i(1 \leqslant j \leqslant n)$, which is implemented by combining each input feature and the single output feature. As a result, the given high-dimensional hypercube $H_i$ can be represented by a set of corresponding decomposed 2D planes.

**Step 2** *Mean value determination for the Menger Curvature of each 2D*: Suppose that a decomposed 2D plane $P_j^i$ contains $h_i$ data instances, the mean of MC of $P_j^i$ can be determined by:

$$MC_j^i = \frac{1}{h_i - 2} \sum_{k=1}^{h_i - 2} mc_k^i, \tag{6}$$

where $mc_k^i$ represents the Menger Curvature value on the $k^{th}$ data point in the 2D plane $P_j^i$, which can be calculated by Eq. 4. Note that the MCs on the two boundary points are not calculable.

**Step 3** *Curvature value determiunation for each hypercube*: The curvature value $C_i$ of corresponding hypercube $H_i$ can be obtained by averaging all decomposed 2D planes as:

$$C_i = \frac{1}{n} \sum_{j=1}^{n} MC_j^i, \tag{7}$$

where $n$ represents the number of features, $MC_j^i$ is the mean of Menger Curvature of $i^{th}$ decomposed 2D plane, which can be obtained by Eq. 6. Note that the dimensionality reduction techniques, such as Principal Component Analysis (PCA), Fuzzy rough feature selection (FRFS) [24], and Linear Discriminant Analysis (LDA) [25], could be applied in this step in helping identify the most relevant features. In this case, a weighted average method can be employed to calculate the curvature of each hypercube. However, this mechanism will remain as a piece of future work.

**Step 4** *Raw Rule Base Generation*:
Based on the obtained curvature values of hypercubes, the important rules for FRI can

be identified. Similar to the original rule base generation approach as detailed in Sec. 2, given a threshold $\phi$, the hypercube with a curvature value higher than the given threshold will be selected to contribute to the generation of a fuzzy rule, which can be expressed as:

$$R_r : \textbf{IF } x_1 \text{ is } A_1^r, x_2 \text{ is } A_2^r, \ldots, x_n \text{ is } A_n^r \textbf{ THEN } y = B_r, \tag{8}$$

where $A_i, i = \{1, \ldots, n\}$, is a triangular fuzzy sets, which can be precisely represented as $A_i = (a_{1i}^r, a_{2i}^r, a_{3i}^r)$, and its extraction procedure is outlined in Fig. 1 and introduced in Sec. 2. The final fuzzy rule base is constructed by comprising all extracted important rules. The generated rule base could be dense or sparse, depends on the given training dataset. In the case of sparse rule base, the T-FRI will be employed to generate inference results.

## 4 Experimentation

The proposed system was evaluated in this section. In particular, two experiments have been carried out, a mathematical model and a real-world application.

### 4.1 Illustrative Example

The problem considered in [19, 20] is re-considered in this work for an illustrative example. This problem is given below:

$$f(x_1, x_2) = \sin\left(\frac{x_1}{\pi}\right) \sin\left(\frac{x_2}{\pi}\right), \tag{9}$$

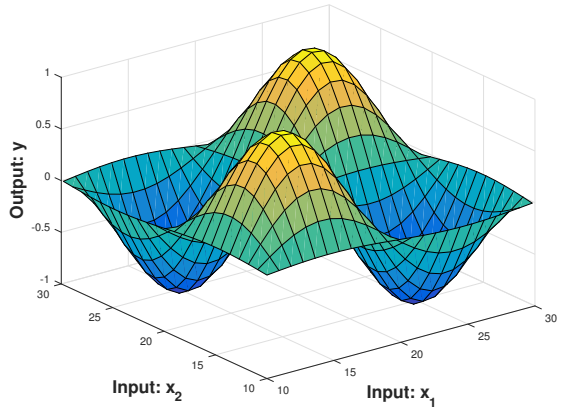
which takes two inputs $x_1$ ($x_1 \in [10, 30]$) and $x_2$ ($x_2 \in [10, 30]$), and produces a single output $y = f(x_1, x_2)$ ($y \in [-1, 1]$), as illustrated in Fig. 3(a).

The problem domain is equally partitioned into 20 grid areas, which consequently results in a total 400 cubes, as shown in Fig. 3(a). Assume that only 10 data instances are contained in $H_1$, as listed in the second, third and fourth columns of Table 1, the curvature values of $H_1$ can be obtained by following steps:

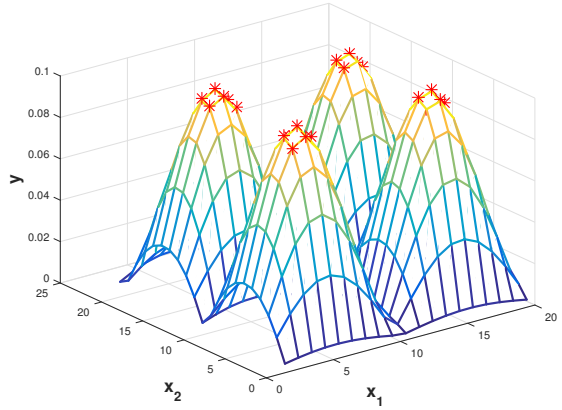
**Step 1**: To break down $H_1$ into two 2D plane $P_1(x_1^1, y^1)$ and $P_2(x_2^1, y^1)$.

**Step 2**: To calculate the menger curvature values for each data point of two decomposed 2D planes by Eq. 4, which are listed in fifth and sixth columns of Table 1, and obtain the mean of menger curvature values of two 2D planes, which are 0.0059 and 0.0059, respectively.

**Step 3**: To determine the curvature value of cube $H_1$ by average two mean of menger curvature values of two 2D planes, which is $C_i = 0.0059$. The curvature values of the rest of cubes can be obtained as the same way, which are listed in Table 2, and also visualised in Fig.3(b).




(a) The problem model 1




(b) Fuzzy sets extraction

Fig. 3: Curvature-based rule base generation

Table 1: Data instances in $H_1$ and corresponding menger curvature value

| No. ($j$) | $x_1^1$ | $x_2^1$ | $y^1$ | $MC_j^1(x_1, y)$ | $MC_j^1(x_2, y)$ |
|---|---|---|---|---|---|
| 1 | 10 | 10 | 0.0017 | N/A | N/A |
| 2 | 10.1 | 10.1 | 0.0054 | 0.0024 | 0.0024 |
| 3 | 10.2 | 10.2 | 0.0110 | 0.0034 | 0.0034 |
| 4 | 10.3 | 10.3 | 0.0187 | 0.0044 | 0.0044 |
| 5 | 10.4 | 10.4 | 0.0282 | 0.0054 | 0.0054 |
| 6 | 10.5 | 10.5 | 0.0397 | 0.0064 | 0.0064 |
| 7 | 10.6 | 10.6 | 0.0531 | 0.0074 | 0.0074 |
| 8 | 10.7 | 10.7 | 0.0683 | 0.0084 | 0.0084 |
| 9 | 10.8 | 10.8 | 0.0852 | 0.0094 | 0.0094 |
| 10 | 10.9 | 10.9 | 0.1038 | N/A | N/A |

Table 2: Curvature values of the cubes

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.0015 | 0.0026 | 0.0035 | 0.0040 | 0.0042 | 0.0039 | 0.0032 | 0.0022 | 0.0010 | 0.0003 | 0.0016 | 0.0027 | 0.0036 | 0.0041 | 0.0041 | 0.0038 | 0.0031 | 0.0020 | 0.0008 | 0.0009 |
| 2 | 0.0123 | 0.0219 | 0.0295 | 0.0342 | 0.0353 | 0.0328 | 0.0269 | 0.0184 | 0.0081 | 0.0029 | 0.0136 | 0.0231 | 0.0303 | 0.0345 | 0.0352 | 0.0322 | 0.0259 | 0.0171 | 0.0067 | 0.0071 |
| 3 | 0.0211 | 0.0382 | 0.0520 | 0.0607 | 0.0628 | 0.0580 | 0.0471 | 0.0318 | 0.0139 | 0.0049 | 0.0235 | 0.0402 | 0.0535 | 0.0614 | 0.0626 | 0.0569 | 0.0453 | 0.0296 | 0.0115 | 0.0124 |
| 4 | 0.0272 | 0.0497 | 0.0686 | 0.0809 | 0.0840 | 0.0771 | 0.0618 | 0.0412 | 0.0179 | 0.0063 | 0.0302 | 0.0525 | 0.0707 | 0.0819 | 0.0837 | 0.0755 | 0.0594 | 0.0383 | 0.0147 | 0.0162 |
| 5 | 0.0305 | 0.0562 | 0.0783 | **0.0930** | **0.0967** | 0.0884 | 0.0703 | 0.0464 | 0.0200 | 0.0071 | 0.0339 | 0.0593 | 0.0807 | **0.0942** | **0.0963** | 0.0865 | 0.0674 | 0.0430 | 0.0165 | 0.0186 |
| 6 | 0.0312 | 0.0577 | 0.0805 | **0.0959** | **0.0998** | **0.0911** | 0.0722 | 0.0476 | 0.0205 | 0.0072 | 0.0348 | 0.0609 | 0.0831 | **0.0971** | **0.0993** | 0.0891 | 0.0693 | 0.0441 | 0.0169 | 0.0196 |
| 7 | 0.0295 | 0.0543 | 0.0754 | 0.0893 | **0.0929** | 0.0850 | 0.0677 | 0.0448 | 0.0194 | 0.0069 | 0.0328 | 0.0573 | 0.0777 | **0.0904** | **0.0925** | 0.0832 | 0.0650 | 0.0416 | 0.0160 | 0.0194 |
| 8 | 0.0252 | 0.0458 | 0.0629 | 0.0739 | 0.0766 | 0.0705 | 0.0568 | 0.0380 | 0.0166 | 0.0059 | 0.0280 | 0.0483 | 0.0647 | 0.0748 | 0.0763 | 0.0691 | 0.0546 | 0.0353 | 0.0137 | 0.0180 |
| 9 | 0.0180 | 0.0323 | 0.0438 | 0.0509 | 0.0527 | 0.0487 | 0.0398 | 0.0270 | 0.0119 | 0.0042 | 0.0199 | 0.0340 | 0.0450 | 0.0515 | 0.0525 | 0.0478 | 0.0383 | 0.0251 | 0.0098 | 0.0152 |
| 10 | 0.0082 | 0.0146 | 0.0196 | 0.0226 | 0.0234 | 0.0217 | 0.0179 | 0.0122 | 0.0054 | 0.0019 | 0.0091 | 0.0154 | 0.0201 | 0.0229 | 0.0233 | 0.0213 | 0.0172 | 0.0114 | 0.0045 | 0.0108 |
| 11 | 0.0029 | 0.0052 | 0.0070 | 0.0081 | 0.0083 | 0.0077 | 0.0064 | 0.0044 | 0.0019 | 0.0007 | 0.0033 | 0.0055 | 0.0072 | 0.0081 | 0.0083 | 0.0076 | 0.0061 | 0.0041 | 0.0016 | 0.0052 |
| 12 | 0.0136 | 0.0243 | 0.0327 | 0.0379 | 0.0392 | 0.0363 | 0.0298 | 0.0203 | 0.0090 | 0.0032 | 0.0150 | 0.0255 | 0.0336 | 0.0383 | 0.0390 | 0.0357 | 0.0287 | 0.0189 | 0.0074 | 0.0012 |
| 13 | 0.0221 | 0.0400 | 0.0545 | 0.0637 | 0.0660 | 0.0609 | 0.0494 | 0.0333 | 0.0146 | 0.0052 | 0.0245 | 0.0421 | 0.0561 | 0.0644 | 0.0657 | 0.0597 | 0.0475 | 0.0309 | 0.0120 | 0.0073 |
| 14 | 0.0278 | 0.0509 | 0.0703 | 0.0830 | 0.0862 | 0.0790 | 0.0633 | 0.0421 | 0.0183 | 0.0065 | 0.0309 | 0.0537 | 0.0724 | 0.0840 | 0.0858 | 0.0774 | 0.0608 | 0.0391 | 0.0151 | 0.0125 |
| 15 | 0.0307 | 0.0566 | 0.0790 | **0.0939** | **0.0977** | 0.0892 | 0.0709 | 0.0468 | 0.0202 | 0.0071 | 0.0342 | 0.0598 | 0.0814 | **0.0951** | **0.0973** | 0.0873 | 0.0680 | 0.0434 | 0.0166 | 0.0163 |
| 16 | 0.0311 | 0.0575 | 0.0803 | **0.0956** | **0.0994** | **0.0908** | 0.0720 | 0.0474 | 0.0204 | 0.0072 | 0.0347 | 0.0607 | 0.0828 | **0.0968** | **0.0990** | 0.0888 | 0.0691 | 0.0440 | 0.0168 | 0.0186 |
| 17 | 0.0291 | 0.0534 | 0.0741 | 0.0878 | **0.0912** | 0.0835 | 0.0667 | 0.0442 | 0.0191 | 0.0068 | 0.0324 | 0.0564 | 0.0764 | 0.0889 | **0.0909** | 0.0818 | 0.0640 | 0.0410 | 0.0158 | 0.0196 |
| 18 | 0.0244 | 0.0443 | 0.0608 | 0.0713 | 0.0739 | 0.0680 | 0.0549 | 0.0368 | 0.0161 | 0.0057 | 0.0271 | 0.0467 | 0.0625 | 0.0721 | 0.0736 | 0.0667 | 0.0528 | 0.0342 | 0.0132 | 0.0194 |
| 19 | 0.0168 | 0.0302 | 0.0409 | 0.0475 | 0.0491 | 0.0455 | 0.0371 | 0.0252 | 0.0111 | 0.0039 | 0.0187 | 0.0318 | 0.0420 | 0.0480 | 0.0489 | 0.0446 | 0.0358 | 0.0235 | 0.0092 | 0.0179 |
| 20 | 0.0068 | 0.0121 | 0.0162 | 0.0187 | 0.0193 | 0.0179 | 0.0148 | 0.0101 | 0.0045 | 0.0016 | 0.0075 | 0.0127 | 0.0166 | 0.0189 | 0.0192 | 0.0176 | 0.0142 | 0.0094 | 0.0037 | 0.0150 |

Given a threshold $\phi = 0.09$, which is identified by human expert knowledge, curvature values of 23 cubes are greater than $\phi$, as shown as bold in Table 2, and also are marked as $*$ (i.e. the selected tubes) in Fig. 3(b). Therefore, those 23 cubes are selected to extract to the fuzzy rules. As a result, the generation rule base comprises with total 23 rules, which exactly same as the rule base generated by profile curvature-based approach provided in [19]. The aim of this illustrative example is to demonstrate the working procedure of the proposed rule base generation method, and its competitive ability will be presented in the next experiment by involving a real-world scenario.

### 4.2 Real-world Scenario

In this experiment, a dataset, which was derived from images in the Mammographic Image Analysis Society (MIAS) database [26] and used for breast cancer risk assessment, is used for evaluation purpose. The dataset includes a set of Medio-Lateral Oblique (MLO) left and right mammogram of 161 women (in total 322 data samples). Each data sample has been pre-processed and represented by 280 features, as well as to be labelled into one of six classes, which are indicated by a proportion of dense breast tissue method in breast cancer risk assessment [27].

In this experiment, all 280 features of the MIAS dataset are selected as the inputs, the six integer numbers $(1 - 6)$, which indicate the six classes, are used as a single output to construct a high-dimensional dataset. The proposed curvature-based rule base generation approach is used to generate a sparse rule base, and the T-FRI inference approach is applied to produce the inference result. The 10-fold cross-validation strategy was employed for system evaluations. In order to evaluate the performance of the proposed approach, eight commonly used classification approaches, including Gaussian Naive Bayes [9], Naive Bayes [9], $k$-Nearest Neighbours ($k$-NN) [28], Multi-functional nearest-neighbour classification (MFNN) [29], Logistic Regression [9], Random Forest [9], Adaptive Boosting (AdaBoost) [9], and Linear Support Vector Machine (Linear SVM) [9], were also implemented with the same experimental setup. Note that the min-max normalisation method was applied for all experimentations for noise reduction. The accuracy of the classification results for each class obtained by different approaches including the proposed one are listed in Fig. 4, which confirmed the competitive ability of the proposed system.

## 5 Conclusion

The existing curvature-based rule base generation approach has been modified using a different curvature calculation approach in this work for more efficient fuzzy interpolation in the setting of handling high-dimensional data, in addition to reducing the space and time complexity with a sparse fashion. Though promising experimental results have been obtained, a faster version of this method is expected as a possible piece of future work, which can be achieved, for example, by investigating the computation of the second-order derivatives in a more efficient way. Besides, it is also worthwhile to investigate how the curvature values can be directly used in helping generate TSK style rule bases.

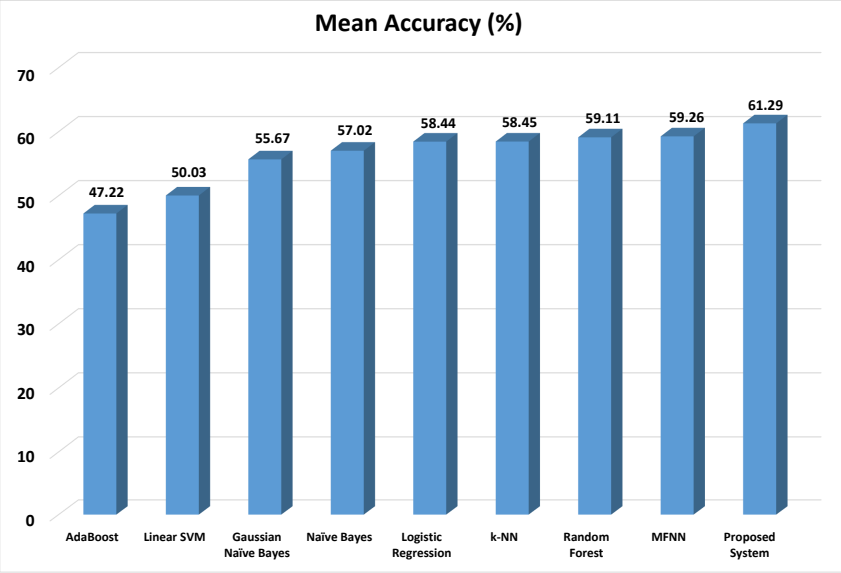



Fig. 4: Experimental results comparison

## References


1. L. T. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *Int. J. Appox. Reasoning*, 9(3):197–225, 1993.
2. Z. Huang and Q. Shen. Fuzzy interpolative reasoning via scale and move transformations. *IEEE Trans. Fuzzy Syst.*, 14(2):340–359, 2006.
3. Z. Huang and Q. Shen. Fuzzy interpolation and extrapolation: A practical approach. *IEEE Trans. Fuzzy Syst.*, 16(1):13–28, 2008.
4. Q. Shen and L.Yang. Generalisation of scale and move transformation-based fuzzy interpolation. *JACIII*, 15(3):288–298, 2011.
5. J. Li, Y. Qu, H. P. H. Shum, and L. Yang. Tsk inference with sparse rule bases. In *Proc. UK Work. Comput. Intell.*, pages 107–123, 2016.
6. L. Yang and Q. Shen. Adaptive fuzzy interpolation. *IEEE Trans. Fuzzy Syst.*, 19(6):1107–1126, 2011.
7. L. Yang and Q. Shen. Closed form fuzzy interpolation. *Fuzzy Sets Syst.*, 225:1–22, 2013.
8. L. Yang, F. Chao, and Q. Shen. Generalized adaptive fuzzy rule interpolation. *IEEE Trans. Fuzzy Syst.*, 25(4):839–853, 2016.
9. Z. Zuo, J. Li, P. Anderson, L. Yang, and N. Naik. Grooming detection using fuzzy-rough feature selection and text classification. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 1–8, 2018.
10. N. Elisa, J. Li, Z. Zuo, and L. Yang. Dendritic cell algorithm with fuzzy inference system for input signal generation. In *Proc. UK Work. Comput. Intell.*, pages 203–214, 2018.

11. Z. Zuo, J. Li, B. Wei, L. Yang, F. Chao, and N. Naik. Adaptive activation function generation for artificial neural networks through fuzzy inference with application in grooming text categorisation. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2019.
12. J. Li, L. Yang, H. P. H. Shum, G. Sexton, and Y. Tan. Intelligent home heating controller using fuzzy rule interpolation. In *Proc. UK Work. Comput. Intell.*, 2015.
13. J. Li, L. Yang, X. Fu, F. Chao, and Y. Qu. Dynamic qos solution for enterprise networks using tsk fuzzy interpolation. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 1–6, 2017.
14. K. Yin, K. Xiang, M. Pang, J. Chen, P. Anderson, and L.Yang. Personalised control of robotic ankle exoskeleton through experience-based adaptive fuzzy inference. *IEEE Access*, 7:72221–72233, 2019.
15. L. Yang, J. Li, F. Chao, P. Hackney, and M. Flanagan. Job shop planning and scheduling for manufacturers with manual operations. *Expert Syst.*, page e12315, 2018.
16. L. T. Koczy and K. Hirota. Size reduction by interpolation in fuzzy rule bases. *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, 27(1):14–25, 1997.
17. J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang. Experience-based rule base generation and adaptation for fuzzy interpolation. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 102–109, 2016.
18. C.-W. Tao. A reduction approach for fuzzy rule bases of fuzzy controllers. *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, 32(5):668–675, 2002.
19. Yao Tan, Hubert P. H. Shum, Fei Chao, V. Vijayakumar, and Longzhi Yang. Curvature-based sparse rule base generation for fuzzy rule interpolation. *Journal of Intelligent and Fuzzy Systems*, 36(5):4201–4214, 2019.
20. Y. Tan, J. Li, M. Wonders, F. Chao, H. P. H. Shum, and L. Yang. Towards sparse rule base generation for fuzzy rule interpolation. In *Proc. IEEE Int. Conf. Fuzzy Syst.*, pages 110–117, 2016.
21. S. D. Peckham. Profile, plan and streamline curvature: A simple derivation and applications. *Proc. Geomorphometry*, 4:27–30, 2011.
22. J. Li, L. Yang, Y. Qu, and G. Sexton. An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation. *Soft Computing*, 22(10):3155–3170, 2018.
23. J.-C. Léger. Menger curvature and rectifiability. *Ann. of Math.*, 149:831–869, 1999.
24. R. Jensen and Q. Shen. New approaches to fuzzy-rough feature selection. *IEEE Trans. Fuzzy Syst.*, 17(4):824–838, 2008.
25. A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien. Linear discriminant analysis: A detailed tutorial. *AI Comms.*, 30(2):169–190, 2017.
26. J. Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis, and I. Ricketts. Mammographic image analysis society (mias) database v1.21. https://www.repository.cam.ac.uk/handle/1810/250394/, 2015.
27. N. F. Boyd, J. W. Byng, R. A. Jong, E. K. Fishell, L. E. Little, A. B. Miller, G. A. Lockwood, D. L. Tritchler, and M. J. Yaffe. Quantitative classification of mammographic densities and breast cancer risk: results from the canadian national breast screening study. *J. Natl. Cancer Inst.*, 87(9):670–675, 1995.
28. David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.
29. Y. Qu, C. Shang, N. Mac Parthaláin, W. Wu, and Q. Shen. Multi-functional nearest-neighbour classification. *Soft Computing*, 22(8):2717–2730, 2018.